

Appl. No. 09/992,642

In the Claims

1-2. (cancelled)

3. (currently amended) A method of enabling dynamic optimization of a computer program, comprising:

5       generating annotation information about said computer program, said annotation information being derived from information held by a compiler about references to individual memory locations; and

10       storing said annotation information with said computer program, said annotation information enabling a dynamic optimizer to optimize said computer program during execution.

15       4. (original) The method of claim 3, wherein generating annotation information comprises generating annotation information enabling replacement of subroutine calls with inline program code in said computer program while said computer program is being executed.

20       5. (original) The method of claim 3, wherein generating annotation information comprises a compiler generating said annotation information.

6. (original) The method of claim 3, wherein said computer program comprises at least one executable file.

7. (original) The method of claim 3, wherein said computer program comprises at least one source file.

25       8. (original) The method of claim 3, wherein said generating annotation information comprises generating

BEST AVAILABLE COPY

Appl. No. 09/992,642

annotation information derived from runtime architecture and software conventions.

9. (cancelled)

10. (original) The method of claim 3, wherein said  
5 generating annotation information comprises generating  
annotation information identifying a unique stack pointer  
register to be used by said computer program.

11. (original) The method of claim 3, wherein said  
generating annotation information comprises generating  
10 annotation information comprising a list of non-ambiguous  
memory locations.

12. (original) The method of claim 11, wherein said  
annotation information enables said dynamic optimizer to  
obtain canonical names for said non-ambiguous memory  
15 locations.

13. (original) The method of claim 11, wherein said  
non-ambiguous memory locations comprise stack frame locations.

14. (original) The method of claim 3, wherein said  
generating annotation information comprises generating  
20 annotation information comprising a mapping of memory  
references to all non-ambiguous locations which are  
referenced.

15. (original) The method of claim 3, wherein said  
generating annotation information comprises generating  
25 annotation information comprising a list of canonical names of  
stack frame locations that are promotable.

Appl. No. 09/992,642

16. (original) The method of claim 3, wherein said generating annotation information comprises generating annotation information comprising a guarantee that no stack frame location is live beyond the scope of the stack frame.

5 17. (original) The method of claim 3, wherein said generating annotation information comprises generating annotation information comprising a format and a location of stack unwinding information.

10 18. (currently amended) A method of dynamically optimizing a computer program, comprising:

reading annotation information derived from runtime architecture and software conventions used to compile said computer program, said annotation information being stored with said computer program; and

15 dynamically optimizing said computer program based on said annotation information while said computer program is being executed.

20 19. (original) The method of claim 18, wherein said dynamically optimizing said computer program comprises a binary translator optimizing said computer program.

20. (original) The method of claim 18, wherein said dynamically optimizing said computer program comprises replacing subroutine calls in said computer program with inline program code.

25 21. (original) The method of claim 18, wherein said dynamically optimizing said computer program comprises removing redundant callee-save register restores.

Appl. No. 09/992,642

22. (original) The method of claim 18, wherein said dynamically optimizing said computer program comprises propagating constant arguments within said computer program.

5 23. (original) The method of claim 18, wherein said dynamically optimizing said computer program comprises promoting local data from a stack frame location to a register.

10 24. (original) The method of claim 18, wherein said dynamically optimizing said computer program comprises removing redundant callee register saves.

25. (original) The method of claim 18, wherein said dynamically optimizing said computer program comprises removing stack frame allocation.

15 26. (currently amended) Apparatus for enabling dynamic optimization of a computer program, the apparatus comprising:  
one or more computer readable storage media; and  
computer executable instructions stored in the one or more computer readable storage media, the computer executable instructions comprising:

20 instructions for generating annotation information about said computer program, wherein said annotation information enables a dynamic optimizer to optimize said computer program during execution, said annotation information being derived from information held by a compiler about references to individual memory locations; and

25 instructions for storing said annotation information with said computer program.

Appl. No. 09/992,642

27. (currently amended) Apparatus for dynamically optimizing a computer program, the apparatus comprising:

one or more computer readable storage media; and  
computer executable instructions stored in the one  
or more computer readable storage media, the computer  
executable instructions comprising:

instructions for reading annotation information  
derived from runtime architecture and software  
conventions used to compile said computer program,  
said annotation information being stored with said  
computer program; and

instructions for dynamically optimizing said  
computer program based on said annotation  
information while said computer program is being  
executed.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**